

Data Analysis (Prof. Ugo Ala) - R Practical N

Cognome Nome, Cognome Nome, Cognome Nome

01-01-0000

Index

Introduction	1
Environment	1
Working directory and Packages	2
Sbobina template	3
Header	3
Introduction	3
Sbobina template	5
Header	5
Introduction	6
Notes and suggestions	7
Title	8
Render	9
Warnings	9

Introduction

Environment

The exercise was run in a Docker, at the moment only a version with JupyterLab is available. The image can be pulled from GitHub:

```
docker pull ghcr.io/maiolino-au/data_analysis:latest
```

To run it open a terminal in the working directory and run this:

- In Windows

```
@echo off
set "CURRENT_DIR=%cd%"
docker run -it --rm -p 8787:8787 -v "%CURRENT_DIR%:/sharedFolder"
  ↪ ghcr.io/maiolino-au/data_analysis:latest
```

- In Linux / MacOS

```
docker run -it --rm -p 8787:8787 -v ./sharedFolder
  ↪ ghcr.io/maiolino-au/data_analysis:latest
```

Repository link: https://github.com/Maiolino-Au/Data_Analysis

The data sets used are available in the repository or on moodle.

The scripts assume that the working directory is the same where the data are stored.

Working directory and Packages

First of all, we need to set the correct working directory: all the commands assume that the datas are stored in the working directory. I worked in a docker container to which i have shared a directory from my PC, called `/sharedFolder` inside the docker. In the directory there is one specific for this lesson, called `Practical_1`. Therefore:

```
setwd("/sharedFolder/Practical_1/")
```

Then we need to load the packages we are going to use. I used `suppressPackageStartupMessages()` to avoid printing all the startup messages of each package.

```
suppressPackageStartupMessages({  
  library(phyloseq)  
  library(dplyr)  
  library(tidyr)  
  library(stringr)  
  library(microbiome)  
  library(microbial)  
  library(vegan)  
  library(usedist)  
  library(ggplot2)  
  library(nortest)  
  library(car)  
})
```

Sbobina template

Here you find the Rmarkdown template for the sbobina and the pdf rendered from it.

Header

Here there are the settings for the file, you need to modify * Names * Date

You can modify * toc_depth: 2 if you want more, or fewer, title levels displayed in the table of content/index

title: "Data Analysis (Prof. Ugo Ala) - R Practical N"

author: "Cognome Nome, Cognome Nome, Cognome Nome"

date: "01-01-0000"

output:

pdf_document:

latex_engine: xelatex

keep_tex: true

toc: true

toc_depth: 2

header-includes:

- \usepackage{fvextra}

- \usepackage{fancyhdr}

- \DefineVerbatimEnvironment{Highlighting}{Verbatim}{breaklines,breakanywhere=true,commandchars=\\\{\}

- \fvset{breaklines=true, breakanywhere=true}

- \renewcommand{\contentsname}{Index}

- \AtBeginDocument{

\pagestyle{fancy}

\fancyhead[L]{}

\fancyhead[R]{}

\fancyhead[C]{Data Analysis - R Practical N - 01-01-0000 - Cognome N., Cognome N., Cognome N.}

\fancyfoot[C]{\thepage}

\renewcommand{\sectionmark}[1]{}

\renewcommand{\subsectionmark}[1]{}

}

\thispagestyle{empty}

\newpage

\pagenumbering{arabic}

\setcounter{page}{1}

Introduction

Leave this section

```
# Introduction
```

```
## Environment
```

The exercise was run in a Docker, at the moment only a version with JupyterLab is
↪ available. The image can be pulled from GitHub:

```
```sh
docker pull ghcr.io/maiolino-au/data_analysis:latest
```
```

To run it open a terminal in the working directory and run this:

* In Windows

```
```powershell
@echo off
set "CURRENT_DIR=%cd%"
docker run -it --rm -p 8787:8787 -v "%CURRENT_DIR%:/sharedFolder"
 ⇨ ghcr.io/maiolino-au/data_analysis:latest
```
```

* In Linux / MacOS

```
```sh
docker run -it --rm -p 8787:8787 -v ./sharedFolder
 ⇨ ghcr.io/maiolino-au/data_analysis:latest
```
```

Repository link: https://github.com/Maiolino-Au/Data_Analysis

The datas used are available in the repository or on moodle.

The scripts assume that the working directory is the same where the data are stored.

\newpage

For the this section * change `setwd("/sharedFolder/Practical_1/")` with the correct directory (remember that `Practical_1` is in the `sharedFolder`, so it is shared to the container from your computer) * add packages if needed

Working directory and Packages

First of all, we need to set the correct working directory: all the commands assume that

- ⇨ the datas are stored in the working directory. I worked in a docker container to
- ⇨ which i have shared a directory from my PC, called `/sharedFolder` inside the docker.
- ⇨ In the directory there is one specific for this lesson, called `Practical_1`.
- ⇨ Therefore:

```
```r
setwd("/sharedFolder/Practical_1/")
```
```

Then we need to load the packages we are going to use. I used

- ⇨ `suppressPackageStartupMessages()` to avoid printing all the startup messages of each
- ⇨ package.

```
```r
```

```
suppressPackageStartupMessages({
 library(phyloseq)
 library(dplyr)
 library(tidyr)
 library(stringr)
 library(microbiome)
 library(microbial)
 library(vegan)
 library(usedist)
 library(ggplot2)
 library(nortest)
 library(car)
})
...

\newpage
```

## Sbobina template

### Header

Here there are the settings for the file, you need to modify \* Names \* Date

You can modify \* toc\_depth: 2 if you want more, or fewer, title levels displayed in the table of content/index

---

title: "Data Analysis (Prof. Ugo Ala) - R Practical N"

author: "Cognome Nome, Cognome Nome, Cognome Nome"

date: "01-01-0000"

output:

pdf\_document:

latex\_engine: xelatex

keep\_tex: true

toc: true

toc\_depth: 2

header-includes:

- \usepackage{fvextra}

- \usepackage{fancyhdr}

- \DefineVerbatimEnvironment{Highlighting}{Verbatim}{breaklines,breakanywhere=true,commandchars=\\\{\}

- \fvset{breaklines=true, breakanywhere=true}

- \renewcommand{\contentsname}{Index}

- \AtBeginDocument{

  \pagestyle{fancy}

  \fancyhead[L]{}

  \fancyhead[R]{}

  \fancyhead[C]{Data Analysis - R Practical N - 01-01-0000 - Cognome N., Cognome N., Cognome N.}

  \fancyfoot[C]{\thepage}

  \renewcommand{\sectionmark}[1]{}

  \renewcommand{\subsectionmark}[1]{}

}

---

\thispagestyle{empty}

\newpage

\pagenumbering{arabic}

\setcounter{page}{1}

## Introduction

Leave this section

### # Introduction

#### ## Environment

The exercise was run in a Docker, at the moment only a version with JupyterLab is available. The image can be pulled from GitHub:

```
```sh
docker pull ghcr.io/maiolino-au/data_analysis:latest
```
```

To run it open a terminal in the working directory and run this:

\* In Windows

```
```powershell
@echo off
set "CURRENT_DIR=%cd%"
docker run -it --rm -p 8787:8787 -v "%CURRENT_DIR%:/sharedFolder"
  ⇨ ghcr.io/maiolino-au/data_analysis:latest
```
```

\* In Linux / MacOS

```
```sh
docker run -it --rm -p 8787:8787 -v ./sharedFolder
  ⇨ ghcr.io/maiolino-au/data_analysis:latest
```
```

Repository link: [https://github.com/Maiolino-Au/Data\\_Analysis](https://github.com/Maiolino-Au/Data_Analysis)

The data sets used are available in the repository or on Moodle.

The scripts assume that the working directory is the same where the data are stored.

\newpage

For this section \* change `setwd("/sharedFolder/Practical_1/")` with the correct directory (remember that `Practical_1` is in the `sharedFolder`, so it is shared to the container from your computer) \* add packages if needed

#### ## Working directory and Packages

First of all, we need to set the correct working directory: all the commands assume that

- the datas are stored in the working directory. I worked in a docker container to
- which i have shared a directory from my PC, called ``/sharedFolder`` inside the docker.
- In the directory there is one specific for this lesson, called ``Practical_1``.
- Therefore:

```
``` r
setwd("/sharedFolder/Practical_1/")
```
```

Then we need to load the packages we are going to use. I used

- ``suppressPackageStartupMessages()`` to avoid printing all the startup messages of each
- package.

```
``` r
suppressPackageStartupMessages({
  library(phyloseq)
  library(dplyr)
  library(tidyr)
  library(stringr)
  library(microbiome)
  library(microbial)
  library(vegan)
  library(usedist)
  library(ggplot2)
  library(nortest)
  library(car)
})
```
```

`\newpage`

## Notes and suggestions

There are up to 6 levels of titles (from # to #####), only level 1 and 2 will be shown in the table of content (Index), you can cahge it with `toc_depth`

Use `\newpage` for ending the page

```
Title
text text text

* lista
* lista

text text text

1. lista numerata
2. lista numerata
1. funziona anche se mettete un numero a caso (nel rendere vedete un 3)
```



bisogna mettere una rigavuota tra due cose  
per far sì che siano

separate

importante

\* per

\* liste (specifico per Rmarkdown, in normal markdown you can omit the empty line right  
↪ before the list)

triplette di backtick per un blocco di codice, singolo backtick per una riga di codice

↪ ``mm <- c("m", "m")`` che potete inserire nel testo. utile per dire "ho usato il comando  
↪ ``paste()`` per unire due stringhe"

```
``` r
mm <- c("m", "m")
paste(mm[1], mm[2], sep = " - ")
```
```

```
```
## [1] "m - m"
```
```

una variabile definita in un blocco si salva nel documento e può essere richiamata in  
↪ blocchi successivi

```
``` r
paste(mm[1], mm[2], sep = " - ")
```
```

```
```
## [1] "m - m"
```
```

## Title

text text text

- lista
- lista

text text text

1. lista numerata
2. lista numerata
3. funziona anche se mettete un numero a caso (nel rendere vedete un 3)

bisogna mettere una rigavuota tra due cose per far sì che siano

separate

importante \* per \* liste (specifico per Rmarkdown, in normal markdown you can omit the empty line right before the list)

triplette di backtick per un blocco di codice, singolo backtick per una riga di codice `mm <- c("m", "m")` che potete inserire nel testo. utile per dire “ho usato il comando `paste()` per unire due stringhe”

```
mm <- c("m", "m")
paste(mm[1], mm[2], sep = " - ")
```

```
[1] "m - m"
```

una variabile definita in un blocco si salva nel documento e può essere richiamata in blocchi successivi

```
paste(mm[1], mm[2], sep = " - ")
```

```
[1] "m - m"
```

## Render

```
rmarkdown::render('/sharedFolder/Practical_1/Data_analysis_R_1.Rmd', output_dir = '/sharedFolder/Practical_1')
```

## Warnings

This gave me an error that made the render fail: it generated a warning containing some invisible characters. you can solve it with `suppressMessages` and `suppressWarnings`

```
wilcox.test(xx$observed ~ xx$Status)
boxplot(xx$observed ~ xx$Status)
```

Solved, the boxplot is still printed.

```
suppressMessages(suppressWarnings({
 wilcox.test(xx$observed ~ xx$Status)
 boxplot(xx$observed ~ xx$Status)
}))
```

A similar problem presented itself with the lines right below: `binwidth` was not specified and the message with which the system notified you that the default values were used caused problems. The previous solution didn't work, so I specified the `binwidth`. You could put a specific number but the resulting dot depends on the scale of the plot so the same value (this kind of plot was used 6 times) resulted in widely different dots, some covering the entire image.

```
ggplot(xx, aes(x = Status, y = observed, fill = Status)) +
 geom_boxplot() +
 geom_dotplot(
 binaxis = "y", stackdir = "center",
 binwidth = diff(range(xx$observed, na.rm = TRUE)) / 30 # sets the width of the
 ↪ dots in the dotplot, put explicitly for reasons regarding Rmarkdown
)
```